DOCUMENT RESUME

ED 227 848 IR 050 139

AUTHOR Domangue, J. C.; Karbowski, S. A.
TITLE Predicting the Readability of FORTRAN Programs.
PUB DATE 82
NOTE 20p.; Paper presented at the Annual Convention of the American Psychological Association (Washington, DC, August 23-27, 1982).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Programs; Data Processing; *Programers; *Programing; *Programing Languages; *Readability Formulas
IDENTIFIERS Debugging (Computers); *FORTRAN Programing Language

ABSTRACT

This paper reports the results of two studies of the readability of FORTRAN programs, i.e., the ease with which a programmer can read and analyze programs already written, particularly in the processes of maintenance and debugging. In the first study, low-level characteristics of 202 FORTRAN programs stored on the general-use UNIX systems at Bell Telephone Laboratories were analyzed to see how strongly they correlated with the number of lines of code in each program, i.e., the number of executable statements. Three measures were found to be relatively independent of lines of code: (1) statement length, defined as the total number of operators and operands divided by the number of executable statements; (2) module redundancy, defined as the degree to which the program uses subroutines and functions properly; and (3) average active variables, defined as the average number of variables that must be kept in mind while reading the program. The effects of these three variables on readability were assessed in the second study. In two different surveys of experienced programmers, they accounted for 60 percent and 72 percent of the variance in rated readability. The results must be considered tentative because the sample sizes involved are small. A list of seven references is provided. (Author/ESR)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED227848

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
- Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

Predicting the Readability of FORTRAN Programs

J. C. Domangue
and
S. A. Karbowski

Bell Telephone Laboratories
6 Corporate Place
Piscataway, New Jersey 08854

Paper presented at the 1982 APA Annual Convention, in Washington, D.C.

IR 010 159

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

G.E. Schindler

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ABSTRACT

This memo reports two studies of the readability of FORTRAN programs. In the first study, low-level characteristics of FORTRAN programs were analyzed to see how strongly they correlated with the number of lines of code. Three measures were found to be relatively independent of lines of code. The effects of those three measures on readability were assessed in the second study. In two different samples, they accounted for 66% and 60% of the variance in rated readability. The results must be considered tentative because the sample sizes involved are small. However, the methods used in these studies hold promise for gaining an understanding of program readability.

The work reported here deals with assessing the readability of computer programs. Readability is only one aspect of software quality, but it is considered by many to be an important characteristic (see, for example, Kernighan and Plauger, 1978). Much of a programmer's time is spent in maintenance and debugging tasks, which entail reading and analyzing code that has already been written. It seems reasonable, then, to think of a program as a document to be read and understood by those who are conversant in the programming language.

The purpose of the work was to investigate the feasibility of using surface features of software as predictors of readability. This approach seemed reasonable in light of the large body of literature on prose readability. Standard readability formulas use low level features like sentence length and word frequency, because such features have been shown to predict performance on comprehension tests and other measures of readability (Klare, 1974). Such an approach might be useful in the assessment of software readability if low-level software features could be found that would meet three requirements.

- -

VISUAL 1

- -

First, they should be measurable in all programs in a target language, to ensure wide usage. Second, they should be derived from a strong theoretical base, so that programmers can understand how to use the measurements to improve their programs. Third, they must be valid predictors of readability, so that programmers can use them to improve their programs.

Many such low-level features suggest themselves as indices of program readability. Some examples are Halstead's software science metrics (Halstead, 1977), and the control flow metric proposed by McCabe (1976) (see Shneiderman, 1980 for other examples).

Many measures are thoroughly documented in the literature, but their relative effects on readability are still unclear. One major stumbling block has been the finding that many measures do not predict program readability any better than a simple count of the lines of code (Curtis, Sheppard, Milliman, Borst, and Love, 1979). The reason for this may lie in the relationship between problem complexity and program length. More complex problems will necessarily lead

to more complex and longer programs. Many software metrics are designed to measure program complexity, so it is not surprising that these measures are related to lines of code. There should be indicators of readability that are not strongly tied to problem complexity and thus to lines of code. Such factors might provide a more promising analysis of software readability.

- -

VISUAL 2

- -

The current report focuses on this problem. Two studies are reported. The first is an analysis of low-level characteristics of FORTRAN programs written in a research and development organization. This analysis was done to determine how strongly each characteristic was related to lines of code. In the second study, three of these characteristics were selected, and their effects on program readability, independent of lines of code, were assessed. For several reasons that will be discussed later, Study II was repeated. The effects of the characteristics on program readability were assessed using programs that were selected from a larger population.

STUDY I

In the first study, FORTRAN programs that were found on two general use UNIX* systems at Bell Telephone Laboratories were analyzed. To be included in the analysis, a program had to compile successfully on the UNIX system FORTRAN77 compiler, and had to have at least 10 executable (non-declaration) statements. A total of 202 programs were analyzed.

- -

VISUAL 3

- -

---

* Trademark of Bell Laboratories.

For each program, the following measures were obtained:

1. Lines of code (number of executable statements; declarations were excluded)

2. Number of variables

3. Halstead's N, program length. Total number of operators and operands.

4. Statement length (Halstead's N ÷ lines of code)

5. Module Redundancy A measure derived from Halstead's work. It reflects the degree to which the program uses subroutines and functions properly.

6. Average active variables. A measure that reflects the average number of variables that must be kept in mind while reading the program.

- -

TABLE 1

- -

In the set of 202 programs, lines of code varied from 10 to 1149. Table 1 shows the intercorrelations of the six variables that were observed. Number of variables was strongly related to lines of code ($r = .84$) as was Halstead's length metric, N ($r = .98$). Statement length, module redundancy, and average active variables were all relatively orthogonal to lines of code, with correlations of .05, .28, and .34, respectively. In addition, the intercorrelations of these three variables were low to moderate (.19 to .51).

Thus, three variables were found to be reasonably independent of lines of code and of each other in real world programs. In the second study, the validity of these three factors as indicators of program readability was investigated.

STUDY II

In this second study experienced FORTRAN programmers rated the readability of eight samples of FORTRAN code. These eight programs were chosen from a subset of the programs analyzed in the first study. All programs from one general use system with between 25 and 110 lines of executable code

were included in the subset. The subset contained 50 programs, which were then divided into eight cells according to whether a program was above or below the median value for each of the three independent variables of interest. One program was chosen from each cell. An attempt was made to include programs from as many different authors as possible, and to cover a wide range of each predictor variable. Otherwise, selection was random.

Copies of these eight programs were sent to 31 persons identified as having FORTRAN files in their directories. They were asked to rate the readability of the programs on a seven point scale. They were also asked to rate how well they understood the program, and how easy it would be to debug the programs. Nine subjects returned completed questionnaires.

Responses to the questions on understanding and debugging were strongly related to readability ratings. The intercorrelations of these three variable were all above .90. Because of this only the results from rated readability will be presented.

- -

TABLE 2

- -

Multiple correlation analysis was used to assess the importance of each of the three independent variables and lines of code to average rated readability. The correlations of each of the independent variables and lines of code are shown in Table 2. The best single predictor was module redundancy ($r = -.736$).

Multiple correlation analysis showed that module redundancy and statement length together accounted for 66% of the variance in average rated readability. Average active variables was moderately correlated with readability ($r = -.388$), but it accounted for no additional variance after the effects of module redundancy and statement length were included. The correlation between lines of code and rated readability was low ($r = .056$). The four variables together accounted for 67% of the variance in readability.

The results of this study seemed promising. Variables that are independent of lines of code were found to be strongly related to rated readability of FORTRAN code. However, it did not seem prudent to generalize immediately from results based on a sample of eight programs drawn from a population

of 50. Such a small sample size restricts the representativeness of the sample.

- -

TABLE 3

- -

This is illustrated in Table 3 by the discrepancy in the correlations between statement length and module redundancy that were obtained in the sample (r = -.27) and in the population (r = .07). A larger discrepancy can also be noted between the correlations of statement length and active variables. In the sample the correlation is -.13, while in the population it is .41. Thus, there is some doubt about the sample of programs being representative of the population. Since the number of raters in this study is small (n = 9) the representativeness of the subjects is also questionable.

For these reasons, the study was repeated, but programs were selected from a larger base, that is, from two UNIX systems. Programs with 25 to 110 lines of executable code were divided into eight cells, and one program was selected from each cell using the same criteria as before.

These programs were sent to a new group of 41 persons identified as having FORTRAN files in their directories. They were also asked to rate readability, understandability, and ease of debugging for each program. Thirteen subjects returned completed questionnaires.

As in the first sample, multiple correlation analysis was used to assess the importance of each of the three independent variables and lines of code to average rated readability. The resulting correlations are shown in Table 4.

- -

TABLE 4

- -

The second sample produced results very different from the first. The best single predictor was statement length (r = -.555), and statement length and active variables together accounted for 60% of the variance. Module redundancy and lines of code were only slightly correlated with rated readability (r = .150 and r = .116, respectively). The four

variables together accounted for 88% of the variance, and with lines of code deleted, the predictors accounted for 72% of the variance.

The discrepancy in results between the two samples suggests that sample representativeness may be a problem. It was noted earlier that the intercorrelation of predictor variables was not the same in the first sample as in the total population of programs.

- -

TABLE 5

- -

Table 5 shows the intercorrelations of the predictors in the second sample and the same intercorrelations in the population. Discrepancies show up in the correlation between statement length and lines of code, and the correlation between active variables and lines of code.

The inability to replicate the results in two different samples prevents us from drawing strong conclusions from the data. A more prudent approach would be to repeat the study on a larger scale. If a larger, more representative sample of programs were used with a larger group of subjects, we would have a better chance of determining the real utility of the predictors. However, the overall methods used in the current studies appear to be useful in understanding the readability of computer programs, and we plan to continue to explore these possiblities.

REFERENCES

Anastasi, A. Psychological testing. Macmillan; New York, 1976.

Curtis, B., Sheppard, S. B., Milliman, P., Borst, M. A., and Lane T. Measuring the psychological complexity of software maintenance tasks with the Halstead and McCabe metrics. IEEE Transactions on Software Engineering, 1979, SE-5, 2., 96-104.

Halstead, M. Elements of software science. Elsevier North Holland, Inc., New York, 1977.

Kernighan, B. W., and Plauger, P. J. The elements of programming style. McGraw-Hill; New York, 1978.

Klare, G. Assessing readability. Reading Research Quarterly, 1974, 10, 62-102.

McCabe, T. A complexity measure. IEEE Transactions on Software Engineering, 1976, SE-2, 6, 308-320.

Shneiderman B. Software psychology Winthrop. Cambridge, Mass., 1980.

PREDICTING THE READABILITY OF FORTRAN PROGRAMS

J. C. Domangue
S. A. Karbowski
Bell Telephone Laboratories
8/24/32

VISUAL 1

REQUIREMENTS FOR READABILITY PREDICTORS

1. MEASURABLE IN ALL PROGRAMS IN A TARGET LANGUAGE.

2. DERIVED FROM A STRONG THEORETICAL BACKGROUND.

3. VALID PREDICTORS OF READABILITY.

VISUAL 2

STUDY I	PURPOSE:	TO DETERMINE THE STRENGTH OF THE RELATIONSHIPS BETWEEN SELECTED LOW-LEVEL CHARACTERISTICS OF FORTRAN PROGRAMS AND LINES OF CODE
STUDY II	PURPOSE:	TO ASSESS THE EFFECTS OF THREE LOW-LEVEL CHARACTERISTICS ON THE READABILITY OF COMPUTER PROGRAMS
REPEAT OF STUDY II	PURPOSE:	TO ASSESS THE EFFECTS OF THE THREE CHARACTERISTICS ON THE READABILITY OF PROGRAMS DRAWN FROM A LARGER POPULATION.

VISUAL 3

MEASURES

1. LINES OF CODE (NUMBER OF EXECUTABLE STATEMENTS; DECLARATIONS EXCLUDED
2. NUMBER OF VARIABLES
3. HALSTEAD'S N (PROGRAM LENGTH. TOTAL NUMBER OF OPERATORS AND OPERANDS)
4. STATEMENT LENGTH (HALSTEAD'S N ÷ LINES OF CODE)
5. MODULE REDUNDANCY ($(N - \hat{N}) \div \hat{N}$)
6. AVERAGE ACTIVE VARIABLES

TABLE 1

INTERCORRELATIONS OF SIX VARIABLES
IN A SAMPLE OF 202 FORTRAN FILES

	Variables	Halstead's N	Statement Length	Module Redundancy	Average Active Variables
Lines of Code	.84	.93	.05	.28	.34
Variables		.91	.18	.08	.30
Halstead's N			.15	.28	.40
Statement Length				.19	.41
Module Redundancy					.51

TABLE 2

CORRELATIONS WITH AVERAGE READABILITY IN THE FIRST SAMPLE OF EIGHT PROGRAMS

Four Predictors	$R^2 = .67$
Three Predictors (Module Redundancy, Statement Length, and Active Variables)	$R^2 = .66$
Module Redundancy and Statement Length only	$R^2 = .66$
Module Redundancy	$R = -.736$
Statement Length	$R = -.133$
Active Variables	$R = -.388$
Lines of Code	$R = .056$

TABLE 3

CORRELATION OF VARIABLES WITHIN THE FIRST SAMPLE OF EIGHT PROGRAMS*

	Statement Length	Module Redundancy	Active Variables	Lines of Code
Statement Length	1.00			
Module Redundancy	-.27 (.07)	1.00		
Active Variables	-.13 (.41)	.49 (.51)	1.00	
Lines of Code	.08 (.05)	-.21 (.28)	.42 (.34)	1.00

* (Population correlations in parentheses)

TABLE 4

CORRELATIONS WITH AVERAGE READABILITY IN THE SECOND SAMPLE OF EIGHT PROGRAMS

Four Predictors	$R^2 = .88$
Three Predictors (Module Redundancy, Statement Length, and Active Variables)	$R^2 = .72$
Statement Length and Active Variables only	$R^2 = .60$
Module Redundancy	$R = .150$
Statement Length	$R = -.555$
Active Variables	$R = -.324$
Lines of Code	$R = .116$

TABLE 5

CORRELATION OF VARIABLES WITHIN THE SECOND SAMPLE OF EIGHT PROGRAMS*

	Statement Length	Module Redundancy	Active Variables	Lines of Code
Statement Length	1.00			
Module Redundancy	-.39 (.07)	1.00		
Active Variables	.33 (.41)	.28 (.51)	1.00	
Lines of code	.39 (.05)	-.25 (.28)	.08 (.34)	1.00

* (Population correlations in parentheses)